REMARKS

The Office action has been carefully considered. The Office action rejected claim 15 under 35 U.S.C. § 112, second paragraph as being indefinite because of the use of a trade name. Further, the Office action rejected claims 1-14 and 16-30 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,604,093 to Etzion et al. ("Etzion") in view of U.S. Patent No. 6,477,585 to Cohen et al. ("Cohen"). Finally, the Office action rejected claim 15 under 35 U.S.C. § 103(a) as being unpatentable over Etzion in view of Cohen and in further view of U.S. Patent No. 6,018,627 to Iyengar et al. ("Iyengar"). Applicants have amended claim 15 to obviate the rejection under §112. Regarding the § 103(a) rejections, applicants respectfully disagree.

By present amendment, claims 1 and 15 have been amended for clarification and not in view of the prior art. Applicants submit that the claims as filed were patentable over the prior art of record, and that the amendments herein are for purposes of clarifying the claims and/or for expediting allowance of the claims and not for reasons related to patentability. Reconsideration is respectfully requested.

Applicants thank the Examiner for the interview held (by telephone) on July 14, 2004. During the interview, the Examiner and applicants' attorney discussed the claims with respect to the prior art. The essence of applicants' position is incorporated in the remarks below.

Prior to discussing reasons why applicants believe that the claims in this application are clearly allowable in view of the teachings of the cited and applied references, a brief description of the present invention is presented.

The present invention is directed to a method and system that uses a trigger engine and infrastructure for event registration and handling. In one embodiment, a switchbox component (of which each trigger engine is a proxy) performs the "fan-in" and "fan-out" of the events consumed and generated by the trigger engine. This includes concentrating multiple similar requests for event notification into a single base event. For example, if a first client requests event notification when a remote file exceeds a certain size and a second client requests event notification when the remote file is deleted, the requests are combined into a single base event request for notification when the file is modified in any manner. The switchbox maintains tables to track which client registered for which type of notification. In this manner, only the base event request is registered remotely, reducing the number of events that need to be communicated to remote servers.

Whenever the base event occurs, the switchbox is notified and then analyzes the information accompanying the base event request to determine which registered clients should get the event notification. For example, if the information indicates that the file changed and the file size specified by the first client was exceeded, the first client is notified of the event, but the second client is not. Moreover, the switchbox is capable of combining events in a complex manner, such as to notify a client only when events A, B, and C have occurred.

Types of events include time events, job events and other events (such as file system events as described above). To this end, a job card may be provided by a client, such that a job scheduler causes the job to launch when the proper events occur. The scheduler launches the job by loading a trigger engine to connect the job, via the switchbox, to a job dispatcher. The job dispatcher runs the processes needed by the job on remote agents. A trigger engine may be attached to the dispatcher and the agents to communicate with the switchbox, for instance, to fire an event when a job is either complete or has failed.

Note that the above description is for example and informational purposes only, and should not be used to interpret the claims, which are discussed below.

Turning to the claims, independent claim 1 recites a system for notifying clients of job-related events of an event source, comprising a first trigger engine configured to register event requests, including a first event request from a first client and a second event request from a second client, and to concentrate the first and second event requests into a base event request, a second trigger engine configured to communicate with the first trigger engine to receive a registration of the base event request at the second trigger engine, and further configured to receive notification of an event of the event source corresponding to the base event, and upon receipt of an event instance corresponding to the base event from the event source, the second trigger engine communicating data indicative of the event instance to the first trigger engine, the first trigger engine configured to determine whether the event instance corresponds to the first event request, and if so, to notify the first client of the event instance, and the first trigger engine

configured to determine whether the event instance corresponds to the second event request, and if so, to notify the second client of the event instance.

The Office action rejected claim 1 as unpatentable over Etzion in view of Cohen. More specifically, the Office action contends that Etzion teaches a system for notifying clients of events of an event source (Col. 17, lines 15-20), comprising: a first trigger engine configured to register event requests (Fig. 2, step 40), including first and second event requests, upon receipt of an event instance (Col. 17, lines 15-20), the first trigger engine configured to determine whether the event instance corresponds to first event request, if so, to notify the first client of the event instance (Col. 17, lines 15-20), and the first trigger engine configured to determine whether the event instance corresponds to the second event request, and if so, to notify the second client of the event instance (Col. 17, lines 15-20).

The Office action acknowledges that Etzion fails to teach a second trigger engine and also states that Etzion fails to teach that the second trigger engine registers multiple event requests that are grouped with similar requests into a base request. Applicants are confused in that this language is not recited in claim 1. Nevertheless, the Office action contends that Cohen does teach this random language, namely that Cohen teaches organizing event requests by events types and then utilizing an event management service to determine the user. The Office action, however, is silent with respect to some recitations of claim 1, namely that the second trigger engine is configured to communicate with the first trigger engine to receive a registration of the base event request at the second trigger engine, and further configured to receive notification of an event of the event source

corresponding to the base event; and upon receipt of an event instance corresponding to the base event from the event source, the second trigger engine providing communicating data indicative of the event instance to the first trigger engine. Notwithstanding this recited language that has been ignored, the Office action concludes that it would have been obvious to a person skilled in the art at the time the invention was made to modify Etzion with grouping of event requests as taught by Cohen because grouping event requests allows more efficient organization of the event requests. Applicants respectfully disagree.

By law, in order to establish *prima facie* obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In addition, "all words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). Further, if prior art, in any material respect teaches away from the claimed invention, the art cannot be used to support an obviousness rejection. *In re Geisler*, 116 F.3d 1465, 1471, 43 USPQ2d 1362, 1366 (Fed Cir. 1997). As discussed in greater detail below, the claims of the present invention are thus clearly patentable over the teachings of the cited and applied references as a matter of law.

Etzion is directed, generally, toward an event management system that is able to monitor a number of events such that a particular combination or temporal relationship of certain parameters of these events will result in a notification sent to a user. See column 17, lines 15-20 of Etzion. That is, the system taught by Etzion is a single-tiered event management system wherein a user may be notified by a

single trigger when certain event parameters are met. As such, the platform hosting the event management system utilizes its resources to determine whether some, all, or none of the parameters of a given event have occurred. Upon determining that each parameter has been met, a notification is sent to a user. For example, a user may wish to be informed of a file change that is both over ten kilobytes in size and executed during business hours.

The system may include multiple parameters for multiple events and may notify multiple users, but, as correctly acknowledged in the Office action, nowhere in Etzion is there any cognizance, let alone any suggestion, of using two distinct event triggers in conjunction with each other for the purpose of notifying a single user of an event.

Similarly, Cohen is directed, generally, toward an event management system that utilizes a filter mechanism to communicate event occurrences from multiple sources to multiple users. As such, a user may configure several individual event filters that form an event schema such that the event schema defines the particular events in which the user wishes to be notified. For example, a user may wish to set up a filter that only allows particular events to be forwarded, such as, changes to a particular file made by a specific individual. Once again, Cohen is an example of a single-tiered event management system wherein events are only triggered from a single source before being sent to notification through a filter mechanism.

In contrast, claim 1 is directed to a system that includes a first trigger engine configured to register event requests, including a first event request from a first

client and a second event request from a second client, and a second trigger engine configured to communicate with the first trigger engine for receiving a registration of the base event request at the second trigger engine. That is, the claim 1 is directed to a two-tiered event management system in that the first trigger engine monitors events and, in turn, communicates occurrences of the monitored events to the second trigger engine which may then determines which events will be communicated back to the first trigger engine prior to being sent as a notification to a user. Neither Etzion nor Cohen, whether considered alone or in any permissible combination, teaches or suggests the concept of two event trigger engines communicating with each other to form a two-tiered event management system. At best, the filter mechanism taught by Cohen may be configured to concurrently filter event requests based on multiple parameters (for example both particular file changes, and changes made by a specific individual), however, the filter mechanism of Cohen cannot possibly be construed to be both a first trigger engine and a second trigger engine wherein each trigger engine is able to communicate back and forth using an event communication protocol. To equate such a filter mechanism as being two distinct entities is roughly equivalent to suggesting that an electric power cord comprises a first half and a second half wherein when electric power is delivered to a device, electricity is communicated from the first half of the power cord coupled to a wall outlet to the second half of the power cord coupled to a vacuum cleaner. In essence, the filter mechanism, as taught by Cohen, cannot communicate with itself such that the teachings of Cohen render obvious the recitations of claim 1.

Furthermore, claim 1 recites that the first trigger concentrates the first and second event requests into a base event request and a second trigger engine configured to communicate with the first trigger engine for receiving a registration of the base event request at the second trigger engine. Nowhere in the prior art of record is there any teaching or suggestion of a base event request as recited in claim 1. A base event request is a concentration of two or more event requests to form a base event that is triggered when any of the component event requests are met. That is, the base event request is triggered when any of the component event requests occur. Thus, the base event request is triggered at the first trigger engine, communicated to the second trigger engine, and subsequently analyzed to determining if the second trigger engine should be triggered by the base event at which time another event communication is passed back to the first trigger engine and eventually to an end user. The prior art merely teaches multiple parameters of a single event wherein each parameter must be met in order to trigger the event (Etzion) and multiple single events grouped together to form an event schema such that a user may be notified when one or more events in the schema have occurred (Cohen). The prior art of record simply does not teach or suggest a base event request as recited in claim 1.

Further yet, the Office action contends that it would have been obvious to a person skilled in the art at the time the invention was made to combine the teachings of Etzion and Cohen because grouping event requests allows for more efficient organization of event requests. Thus, the Office action is stating that a better organization of event requests is a motivation to combine the teachings of

Etzion and Cohen which is akin to stating that any improvements to a gasoline engine would be obvious simply because better fuel efficiency is always desired. Such broad conclusory statements do not come close to adequately addressing the issue of motivation to combine, are not evidence of obviousness, and therefore are improper as a matter of law. *In re Dembiczak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999). Further, if anything, the references teach away from such a combination, as discussed above, because each system functions independently using only a single tier of one or more trigger engines, none of which communicate with other.

For at least the foregoing reasons, applicants submit that claim 1 is allowable over the prior art of record.

Applicants respectfully submit that dependent claims 2-18, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 1 and consequently includes the recitations of independent claim 1. As discussed above, both Etzion and Cohen, whether considered alone or in any permissible combination, fail to disclose or suggest the recitations of claim 1. In addition to the recitations of claim 1 noted above, each of these dependent claims includes additional patentable elements.

For example, claim 3 recites the system of claim 1 wherein the first and second trigger engines are each a proxy of a switchbox component. As discussed above, none of the prior art of record, whether considered alone or in any permissible combination, teaches or suggests a first trigger engine and a second trigger engine as recited in claim 3. Nor does any prior art of record, whether

considered alone or in any permissible combination, teach or suggest a first and second trigger engines proxy to a switchbox. In fact, both Etzion and Cohen specifically teach that the one and only trigger engine (or set of single-tiered trigger engines) is part of the server computer that hosts the event management system. Applicants submit that claim 3 is allowable for at least these additional reasons.

As another example, claim 6 recites the system of claim 1 wherein the first trigger engine is a client of the second trigger engine, and wherein the second trigger engine has a least one data structure associated therewith for determining which client event requests correspond to event instances. Again, none of the prior art of record, whether considered alone or in any permissible combination, teaches or suggests a first trigger engine and a second trigger engine as recited in claim 6. Nor does any prior art of record, whether considered alone or in any permissible combination, teach or suggest a first and second trigger engines having a relationship such as the first trigger engine being a client of the second trigger engine. In fact, both Etzion and Cohen specifically teach that the one and only trigger engine is part of the server computer that hosts the event management system. Applicants submit that claim 6 is allowable for at least these additional reasons.

As a final example, amended claim 15 recites the system of claim 1 wherein the first trigger engine communicates with the second trigger engine via a message queuing service. With respect to claim 15, the Office action rejected this claim as being unpatentable over Etzion in view of Cohen and in further view of Iyengar. Again, none of the prior art of record, whether considered alone or in any

permissible combination, teaches or suggests a first trigger engine and a second trigger engine as recited in claim 15. Nor does any prior art of record, whether considered alone or in any permissible combination, teach or suggest a first and second trigger engines having a relationship such as the first trigger engine communicating with the second trigger engine via a message queuing service. In fact, both Etzion and Cohen specifically teach that the one and only trigger engine is part of the server computer that hosts the event management system.

Turning to the next independent claim, claim 19 recites in a computer network, a method for notifying clients of events, comprising receiving from a first client a first request corresponding to a first event on a remote server, the first request including information specific thereto, receiving from a second client a second request corresponding to a second event on the remote server, the second request including information specific thereto, maintaining information specific to each event request in association with each client, concentrating the first and second event requests into a base event request, registering the base event request at the remote server, receiving notification of the base event, the notification including event-specific information about the base event, analyzing the event-specific information, notifying the first client if the event-specific information corresponds to the information specific to the first event request associated with the first client, and notifying the second client if the event-specific information corresponds to the information specific to the second event request associated with the second client.

The Office action rejected claim 19 as being unpatentable over Etzion in view of Cohen. In particular, the Office action stated that claim 19 is directed to a method similar to the system of claim 1 and, as such, is rejected for the same reasons detailed above with respect to claim 1. Applicants respectfully disagree and submit that claim 19 should be examined on its own merits.

Claim 19 is directed to a method that includes, among other things, concentrating the first and second event requests into a base event request, registering the base event request at the remote server, receiving notification of the base event, the notification including event-specific information about the base event, analyzing the event-specific information. Clearly, there is no mention or even any cognizance of the concept of a base event request as recited in claim 19. As used in claim 19, a base event request is a concentration of two or more event requests to form an event that is triggered when any of the component event requests are met. That is, the base event request is triggered when any of the component event requests occur. Thus, the base event request is triggered at the first trigger engine, communicated to the second trigger engine, and subsequently analyzed to determining if the second trigger engine should be triggered by the event at which time another event communication is passed back to the first trigger engine and eventually to an end user. The prior art merely teaches multiple parameters of a single event wherein each parameter must be met to trigger the event (Etzion) and multiple single events grouped together to form an event schema such that a user may be notified when one or more events in the schema